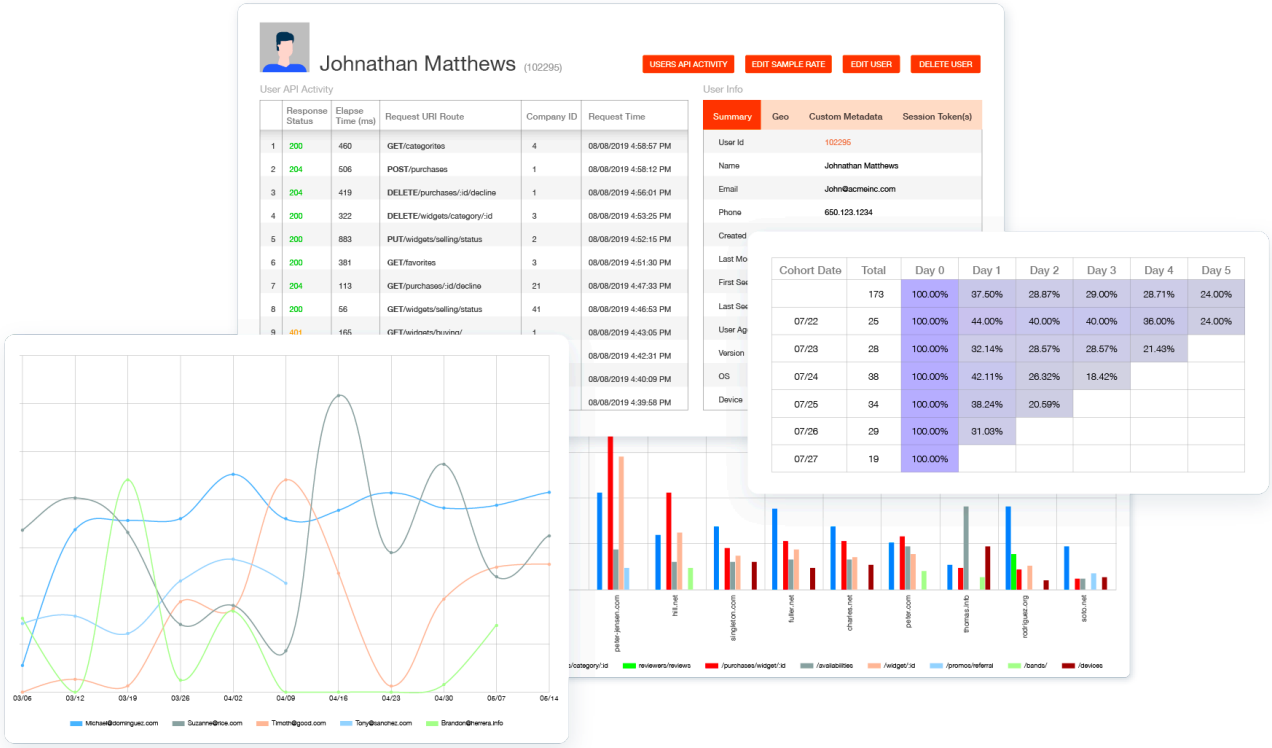
moesif

API Analytics:
The Ultimate Guide to Grow Your Platform Business

Best practices for measuring, tracking, and growing key API and business metrics



APIs are enabling entirely new business models such as API as a Product, developer platforms and ecosystems, as well as new partner opportunities. However, if you want to out-innovate your competition and quickly grow your platform, you will need the right data to make informed decisions.

API Analytics are not just valuable for engineering teams, but across the organization including product owners, customer success, marketing, and sales. At the same time, there is a lot of confusion on what really is considered API Analytics and what are the best practices. This guide is designed to address this confusion.

Copyright © 2020 moesif.inc

Contents

About Us moesif

Moesif is the most advanced API analytics platform used by Thousands of platforms to understand what your most loyal customers are doing with your APIs, how they're accessing them, and from where. Moesif focuses on analyzing real customer data vs just synthetic tests to ensure you're building the best possible API platform for your customers.

What is API Analytics?

API Analytics and reporting includes both engineering focused metrics such as performance and uptime, but also tracking customer and product metrics such as engagement, retention, and developer conversion. There are a variety of methods to perform such analysis which includes basic SQL and Excel to purpose built API analytics platforms.

Copyright © 2020 moesif.inc

Defining Key API Metrics

Each team needs to track different KPIs when it comes to APIs. The API metrics important to infrastructure teams will be different than what API metrics are important to API product or API platform teams. Metrics can also be dependent on where the API is in the product life cycle. An API recently launched will focus more on improving design and usage while sacrificing reliability and backwards compatibility. A team that maintains an API that's been widely adopted by enterprise teams may focus more on driving additional feature adoption per account and give precedence to reliability and backwards compatibility over design.

01 Weekly Active Tokens (WAT)

To measure the success of a platform, we should look at the usage. This can be from the various APIs and SDKs that you have released for third party developers to interact with your platform. The mere act of generating an API key does not measure usage since even non-developers can view API keys without any real use for them. Thus we turn to Weekly Active API Tokens. Since most APIs limit access to authenticated users, we are able to track how many distinct tokens are accessing our API platform on a given week.

Since a single customer can create multiple API keys with various scopes and expiration dates, we can further narrow this down to Weekly Active Integrated Companies. Everyone on your developer relations team should be looking at this metric when deciding on where to invest more time. In order to do so, you should have the instrumentation in place to tie metrics like tokens to developer demographic info such as any marketing channels that brought the developer to your site, company affiliation and size, and even social info such as GitHub stars. With this info, you can break down WAT to find what is contributing the most to your north star metric

02 API usage growth

For many product managers, API usage (along with unique consumers) is the gold standard to measure API adoption. An API should not be just error free, but growing month over month. Unlike requests per minute, API usage should be measured in longer intervals like days or months to understand real trends. If measuring month-over-month API growth, we recommend choosing 28-days instead as it removes any bias due to weekend vs weekday usage and also differences in number of days per month. For example, February may have only 28 days whereas the month before has a full 31 days causing February to appear to have lower usage.

Copyright © 2020 moesif.inc

03 Unique API consumers

Because a month's increase in API usage may be attributed to just a single customer account, it's important to measure API MAU (Monthly Active Users) or unique consumers of an API. This metric can give you an overall health of new customer acquisition and growth. Many API platform teams correlate API MAU to their web MAU, to get a full product health. If web MAU is growing far faster than API MAU, then this could imply a leaky funnel during integration or implementation of a new solution. This is especially true when the core product of the company is an API such as for many B2B/SaaS companies. On the other hand, API MAU can be correlated to API usage to understand where that increased API usage came from (New vs. existing customers). Tools like Moesif can track both individual users calling and API and also link them to companies or organizations.

04 Top customers by API usage

For any company with a focus on B2B, tracking the top API consumers can give you a huge advantage when it comes to understanding how your API is used and where upsell opportunities exist. Many experienced product leaders know that many products exhibit power law dynamics with a handful of power users having a disproportionate amount of usage compared to everyone else. Not surprisingly, these are the same power users that generally bring your company the most revenue and organic referrals.
This means its critical to track what your top 10 customers are actually doing with your API. You can further break this down by what endpoints they are calling and how they're calling them. Do they use a specific endpoint much more than your non-power users? Maybe they found their ah ha moment with your API.

05 API retention

Should you spend more money on your product and engineering or put more money into growth? Retentionand churn (the opposite of retention) can tell you which path to take. A product with high product retention is closer to product market fit than a product with a churn issue. Unlike subscription retention, product retention tracks the actual usage of a product such as an API. While the two are correlated, they are not the same. In general, product churn is a leading indicator of subscription churn since customers who don't find value in an API may be stuck with a yearly contract while not actively using the API. API retention should be higher than web retention as web retention will include customers who logged in, but didn't necessarily integrate the platform yet. Whereas API retention looks at post-integrated customers.

Copyright © 2020 moesif.inc

06 Time to First Hello World (TTFHW)

TTFHW is an important KPI for not just tracking your API product health, but your overall developer experience aka DX. Especially if your API is an open platform attracting 3rd party developers and partners, you want to ensure they are able to get up and running as soon as possible to their first ah ha moment. TTFHW measures how long it takes from first visit to your landing page to an MVP integration that makes the first transaction through your API platform. This is a cross-functional metric tracking marketing, documentation and tutorials, to the API itself.

07 API Calls per business transaction

While more equals better for many product and business metrics, its important to keep the number of calls per business transactions as low as possible. This metric directly reflects the design of the API. If a new customer has to make 3 different calls and piece the data together, this can mean the API does not have the correct endpoints available. When designing an API, it's important to think in terms of a business transaction or what the customer is trying to achieve rather than just features and endpoints. It may also mean your API is not flexible enough when it comes to filtering and pagination..

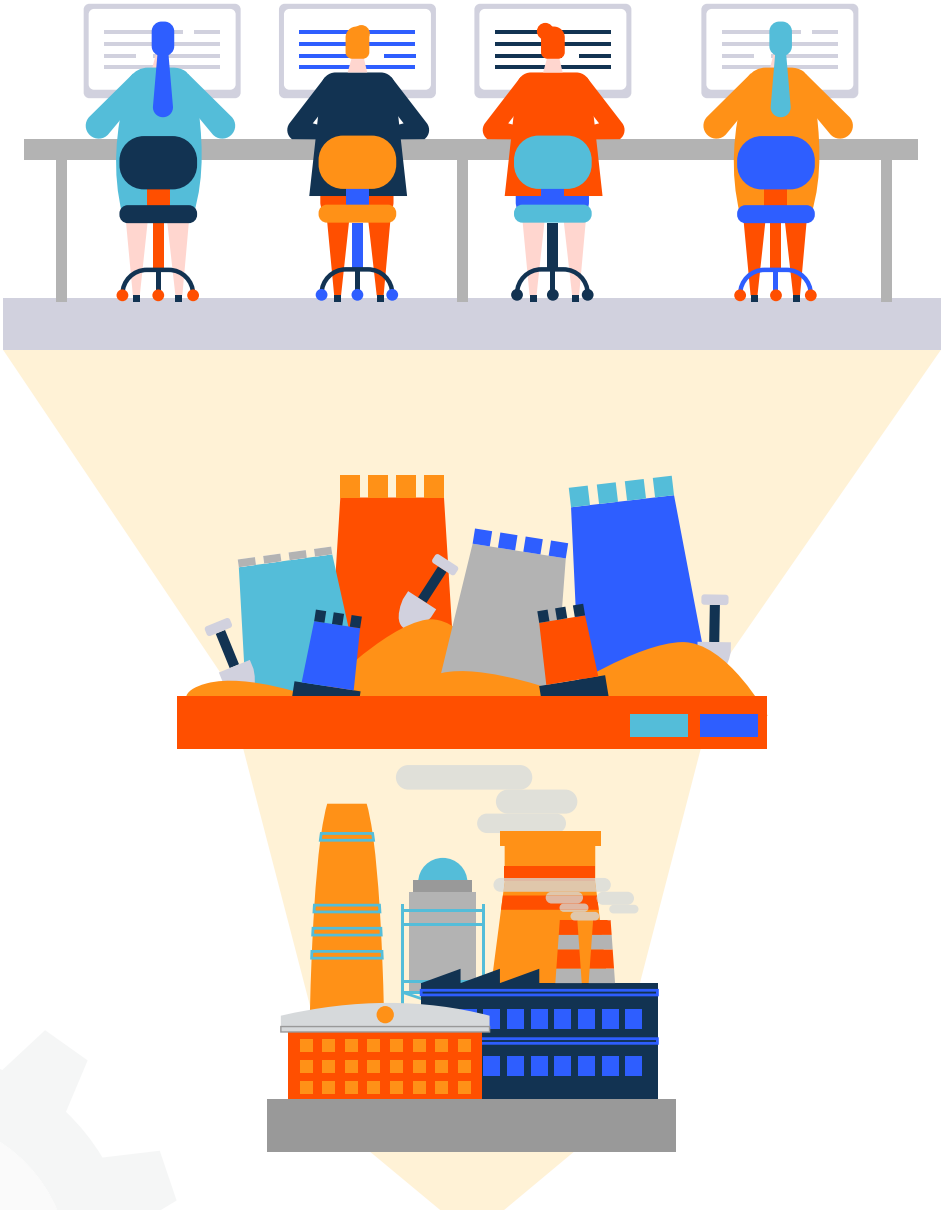
08 SDK and version adoption

Many API platform teams may also have a bunch of SDKs and integrations they maintain. Unlike mobile where you just have iOS and Android as the core mobile operating systems, you may have 10's or even hundreds of SDKs. This can become a maintenance nightmare when rolling out new features. You may selectively roll out critical features to your most popular SDKs whereas less critical features may be rolled out to less popular SDKs. Measuring API or SDK version is also important when it comes to deprecating certain endpoints and features. You wouldn't want to deprecate the endpoint that your highest paying customer is using without some consultation on why they are using it.

For anyone building and working with APIs, its critical to track the correct API metrics. Most companies would not launch a new web or mobile product without having the correct instrumentation for engineering and product. Similarly, you wouldn't want to launch a new API without a way to instrument and track the correct API metrics. Sometimes the KPIs for one team can blend into another team as we saw with the API usage metrics. There can be different ways of looking at the same underlying metric. However, teams should stay focused on looking at the right metrics for their team. For example, product managers shouldn't worry about CPU usage just like infrastructure teams shouldn't worry about API retention. Tools like Moesif API Analytics can help you get started measuring these metrics with just a quick SDK installation.

Copyright © 2020 moesif.inc

The Developer Funnel



You have an API program that developers are adopting, but unsure how much. How long does it take for a new integration to move to revenue generating? If you came from a web or mobile product management background, you may already be familiar with mobile product analytics to measure app engagement and retention. Growing APIs have similar KPIs to measure the success of your API program. This article will talk more about what you should be measuring and how to leverage that information.

While traditional customer funnels will consist of just a marketing and sales funnel component. However, APIs as a product where customers and partners include developers have what is called the developer funnel or integration funnel. The developer funnel is after the marketing funnel and before the sales funnel and has three core stages:

1. Pre-integration Stage
2. Sandbox Stage
3. Production Stage

Copyright © 2020 moesif.inc

Pre-integration stage

Developers enter the pre-integration stage after marketing funnel. While there is a blur for what is considered marketing vs developer funnel, intent is the main factor. Are they just considering a solution through well-defined marketing assets like case studies, content, and demo accounts or committed to testing the solution themselves.

Once the evaluation or consideration phase is positive, there is intent to test. This causes the developer to start the installation which can be seen by creating a blank workspace, downloading a configuration file, or copying an API key. This implies your product is freemium, has a free trial, or other.

Sandbox stage

A potential customer enters the sandbox stage as soon as a single action is taken on your API that gives the developer a warm feeling that they accomplished something and able to verify this is a viable solution for them. The time to go from pre-integration stage to sandbox stage is usually measured as Time to First Hello World (TTFHW). This is a critical KPI that every API product manager should track and reduce as much as possible.

A developer entered the sandbox stage when the integration is successful and there is a clear sign of value creating your aha` moment. These developers start to become an internal advocate for your solution.

Possible reasons for not moving into sandbox stage:

1. Integration issues such as errors and bugs in SDKs or implementation. For example, if the developer received 500 errors from your API or the SDK didn't make any calls to your API. The latter is one of the hardest to measure since the developer completed the implementation but no actions were captured on your side.
2. No easy way to verify that the implementation is correct. If your product looks empty even after integration, a developer may assume it doesn't work. This can happen for security and analytics companies where there are no meaningful insights until a large amount of data is processed.
3. An action was taken on API but unclear what the value is. The action should have clear value and not just a message saying "You're integrated." For a communications API, that value may be sending a single SMS. For an analytics company, it may be capturing and displaying a single event.

Production stage

Once a developer sees value in a solution, the developer wants to push their implementation to production as soon as possible. However, the developer may not be able to push your API to production due to internal blockers, many outside the control of the individual developer.

Your goal is to empower the developer with the necessary assets and tools to eliminate these objections. Once a developer moves into the production stage, there is a measurable amount of traffic going to your API from that account. The time it takes to get to production level traffic can be called Time To First Working App (TTFWA) or Time To First Paid App (TTFPA).

Possible reasons for not moving into production stage:

1. Project priorities. Customer bugs or other project features may take priority. Your communications API integration took a back seat just because the developer had a pile of Jira tickets assigned to him or her.
2. Legal and compliance. Especially in regulated industries such as financial and health, the developer may have to wait until necessary approval from legal.
3. Functional and performance testing. The company may have internal policies for testing third party services which need to be finished first.

Copyright © 2020 moesif.inc

Creating Funnel Goals

In order to have an accurate snapshot of your API business, each funnel boundary requires well-defined goals. Let's use Algolia, a search API, as an example.

01 Pre-integration step goals

The first step for Algolia will be when a developer uses the API key copy feature. At this point, the developer already signed up, and going through the onboarding flow.

02 Sandbox step goals

Algolia has two core features around their API. A query or search API and an index API. Algolia's success metric would be developers who were able to index at least one entity via the index API. If the developer called the query API when no data has been stored in Algolia, that is not considered a success as the query will be empty. In order to keep their integration simple, they can either:

1. Drive users to index a new entity via `POST /1/indexes/` endpoint
2. Use sample data to bootstrap a sample index that the developer can query

Let's say we go with (1), then our funnel conversion point would be any developers who performed at least one `POST /v1/indexes` `operation.` Algolia may want a tighter conversion metric so they may track any developer who performed at least one `POST /v1/indexes` operation AND one `POST /1/indexes/{indexName}/query` operation WHERE response Content-Length > 0. This means the developer successfully indexed at least a single entity and was able to read it back.

03 Production step goals

Creating a metric for what is considered production-level traffic can be hard as it could be industry or project dependent. For Algolia, an e-commerce app has different levels of production use than an internal enterprise app with lots of dynamic content. For the enterprise app with dynamic content, each entity could be reindexed tens or hundreds of times a day programmatically with updated data. However, if the data is never queried, then the value of Algolia hasn't been seen yet. So just looking for developers with more than 1,000 `POST /1/indexes/` operations may not be the best value metric to track. A better value metric might be developers who have more than 1,000 `POST /1/indexes/{indexName}/query` operations WHERE response Content-Length > 0.

Note how we are focusing on the core value of Algolia which is search. We are less focused on auxiliary features like Algolia's monitoring API, reporting API, etc, nor do we look at advanced features like synonyms or rules APIs.

Copyright © 2020 moesif.inc

Developer cohort segmentation

Now that we have an understanding of conversion rates for each stage of the developer funnel, it's important to leverage API Analytics to understand how different cohorts of developers move through your developer funnel. Without it, you can have a false sense of success. For example, you can have a high conversion rate from Pre-integration to Production stage, yet if majority of developers are side projects without buying power, then you may need to reevaluate your targeting. See Figure 1. below for an example developer funnel.

Developer Funnel Chart

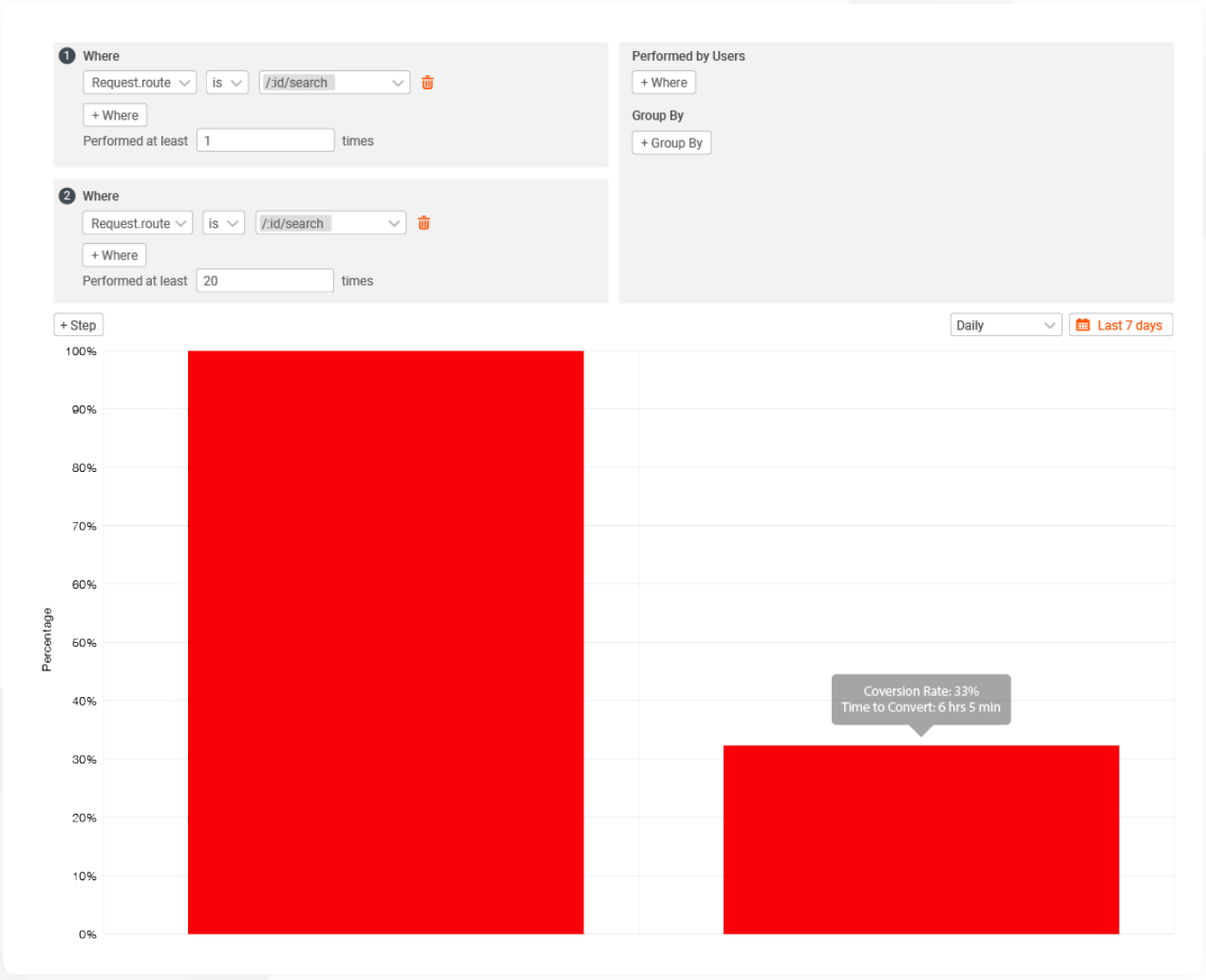


Fig. 1 Developer Funnel Chart

Building Profiles of Users and Companies

In order to segment your funnel further, your API analytics solution should have a way to create User Profiles and Company Profiles.

Once you have this information, your developer funnel analytics become far more valuable. Instead of just tracking the high level integration journey, you're able to see who is integrating and who isn't. For example, we can segment by Company Revenue to compare the conversion rate of users working on side projects vs. the conversion rate of engineers at Fortune 500 companies.

Tracking companies (i.e. accounts) separately from users ensures you're able to create metrics on the number of developers or teams at an organization using your solution.

Recommended variables to track:

User Profile

Email
Name
Job Title
Referring site

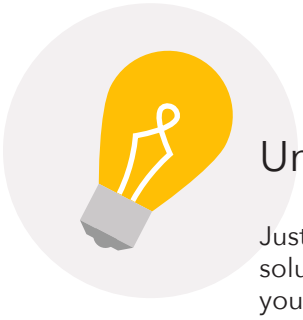
Company Profiles

Company Revenue
Number of Employees
Industry
Alexa Rank
Plan MRR

Comp ID	Company Domain	Number of Employee	City	Country	Alex Ranking	Company Name
460	cooper.info	48000	Jonton	Node.js	3200	Medina PLX
506	smith.com	25000	Bandar	Greece	8600	Moreno, Calson and Prince
419	bell-clark.com	52000	Oneillberg	Portugal	5300	Beck PLC
322	douglas.info	72000	DC	US	5300	Crawford, Maynad and Chelsea
883	smith.info	80000	Trier	Turkey	2200	Drake Schmidt
381	harvey.com	17000	London	England	2500	Andrew, Williams, and Lau
113	ballard.com	70000	Hong Kong	Hong Kong	2877	Paratus
556	brown.com	17000	Kashiwa	Japan	2977	Kyoto Totoro

Fig. 2 List of Company Profiles

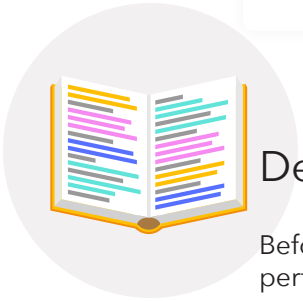
Copyright © 2020 moesif.inc



Understanding API Design

Just like building profiles of users and companies, your API Analytics solution should understand the business functionality of your API. If your API is rest, then tracking just `POST/items/1` and `POST /items/2` would not be valuable. However, adding metadata associated with the API transaction can give you higher level trends on usage patterns. For example, if your API is RESTful, your analytics solution should bucket the API transactions into a specific REST resource operation like `POST/items/:id`. This enables higher level charts such as below which shows the top companies using each REST API operation.

Fig. 3 Segmentation Chart



Defining API engagement

Before we dig into critical business metrics like acquisition channel performance and API growth, we need to define what is considered engagement with an API. After all, a developer just hitting your root end-point or just probing for API health may not be considered real engagement. Continuing with the Algolia example, Algolia's core business value prob is fast search. Thus, their criteria to define engagement on an API is an account performing at least one search `POST /1/indexes/{indexName}/query` in the last 28-days.

Copyright © 2020 moesif.inc

Measuring Developer Acquisition

Efficient new customer acquisition can make or break an API business. It's imperative to consider the conversion rates of our three steps defined in our developer funnel when measuring the effectiveness of online marketing campaigns across traffic sources. API product marketing leaders should move beyond basic metrics like sign ups and consider the developer integration funnel. This enables you to understand how far those sign ups went through the integration steps.

How to properly measure developer marketing channels

If we sync marketing data such as from Urchin Tracking Module (UTM) parameters from marketing automation and analytics tools like Hubspot or Amplitude, then we can leverage this in our cohort analysis. This enables us to take our earlier developer funnel and segment by the channel. Once we do that, we're able to see the conversion rate for each step broken down by campaign.

A channel like Facebook ads may give your B2B solution a large reach, but if a very small fraction of those sign ups end up going through the integration process due to poor targeting, your overall effectiveness of the channel is lowered as it's unlikely for you to extract much value from someone who just signed up without doing anything else.

Distinct API operations/application

If you have an API with hundreds of separate features and entities like GitHub or Intuit's APIs, then measuring how many separate functions each app calls can help you understand the breadth of API usage. It's helpful to plot this as a histogram to find the percentage of power users who are using your API fully compared to other developers who may be using only one or two endpoints. How you track distinct API operations will depend on the architecture of your API.

```
GET /items
GET /items/:id
POST /items

GET /users
GET /users/:id
POST /users

GET /purchases
GET /purchases/:id
POST /purchases
```

REST API with REST Endpoints

```
getItems
createItem
createItem

getUsers
getUser
createUser

getPurchases
getPurchase
createPurchase
```

GraphQL API to track queary and mutate operations

Copyright © 2020 moesif.inc

Measuring API growth and engagement

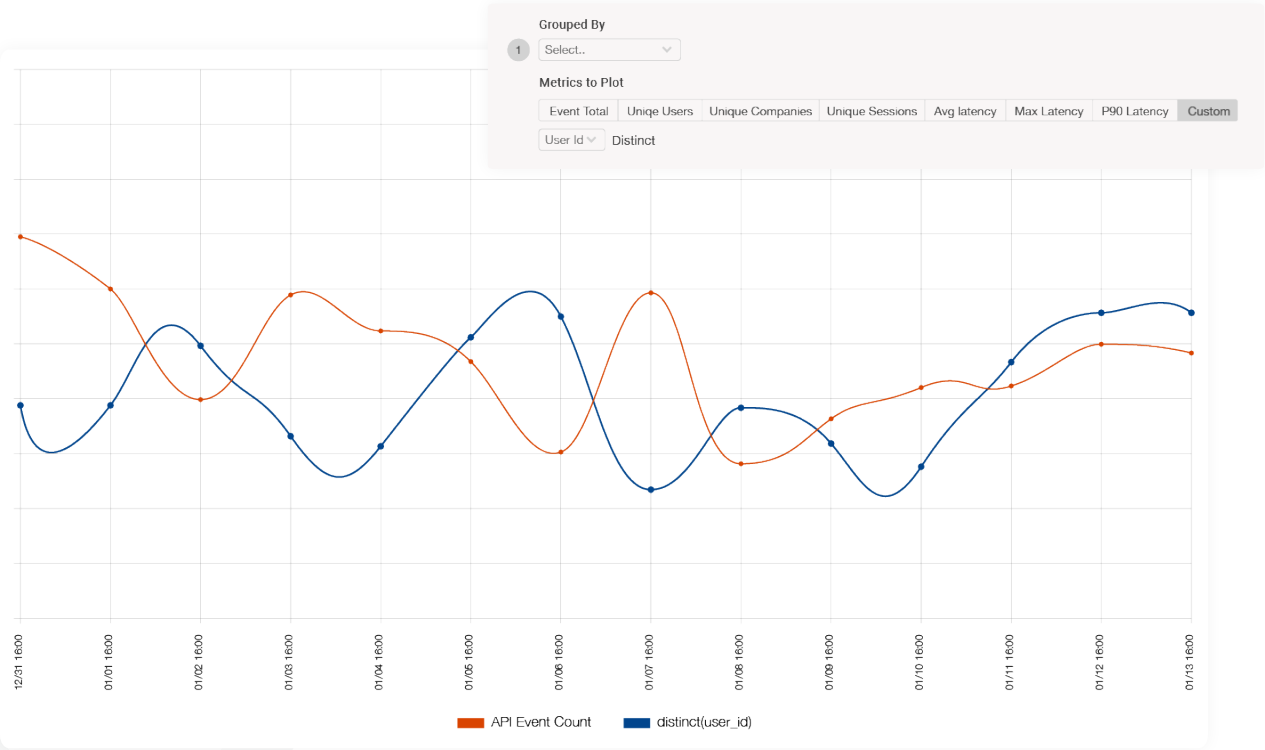


Fig. 4 Time Series chart

API MAU and API DAU

If you want to track growth of your API program, API Monthly Active Users (API MAU) and API Daily Active Users (API DAU) are two KPIs every data-driven product leader should be tracking.

This is similar to web and mobile MAU and can give a high level overview of growth and health. However, instead of tracking distinct users performing UI interactions such as button clicks within your app, API MAU is tracking distinct users performing programmatic interactions with your API. Depending on your industry, a distinct user can be an individual user or developer or it can be distinct companies or accounts. API Analytics products like Moesif can track which is critical for account-level analytics.

28-day API usage/active companies

You should also be tracking total API usage per active company for the same engagement definition. API usage is simply the total API usage in sub of API calls rather than counting distinct users. By dividing API usage by active companies, you can track how each account is growing with your API over time. This metric is a leading indicator for revenue expansion. If API Usage/Active companies are flat (or worse decreasing), your API is not growing with the accounts and may be hard to drive upsells through plan limits. Whereas an API program with a rapidly growing API usage/active companies will enjoy a growing average contract value and top line revenue. Of course, this assumes your pricing is based on some usage metric such as search operations or index operations in Algolia's case.

28-days?

Even though we say MAU, we should really be measuring activity in fixed 28-day windows, not the entire month. By using 28-days, we are able to ignore bias due to the number of days in a month not being consistent. In addition, 28-days ensures each time window starts on the same weekday. This guarantees we always have the same number of weekdays and weekends in each time window. Many companies, especially B2B, have different levels of usage on the weekends compared to weekdays. By keeping these numbers consistent, we can remove those biases.

Copyright © 2020 moesif.inc

Cohort Retention Analysis

There are few metrics more critical than retention for a platform business. If you're acquiring customers for $25, but they stop using your API after a month, then you have a leaky boat. Don't spend more money on developer acquisition until retention is fixed. This requires accurate measurement of API retention.

If you came from a web or mobile product background, you may already be familiar with mobile retention to measure how many acquired users keep using a mobile app. Growing a B2B platform requires tracking similar KPIs to measure the success of your acquisition and product strategies. This article will dig into the best practices for tracking and increasing API retention.

What is API retention?

Retention measures the percent of users within a cohort that return and stay active with your product. What is considered active for your product depends on the type of product. For a streaming mobile app, being active on a day may mean playing a song. For a payments API, it could be processing a credit card payment on a day.

To measure retention accurately, you need to segment your users into cohorts. Usually this is done by sign up date, but for APIs it's recommended to segment based on integration date or Time of First Hello World. This is the date when a user first integrated and made their first transaction through your API. Then, every day, week, or month later we count the number of unique users who returned and performed an action which you consider a strong indicator for being active.

Copyright © 2020 moesif.inc

What does a Retention Chart look like?

The below chart shows new users segmented by day of first API integration. From there, we track the percent of users that are still active on our API each day after. This chart shows 37.5% of new users are still actively making API calls the first day after integration. Five days after integration, only 24% of our initial cohort was still active.

While only 24% of cohort is active 5 days after integration doesn't seem high, this chart can be quite good depending on your use case. Retention should not measure engagement or stickiness levels. Rather, you should use retention curves to see how many users are retained. The best way to to tell is how flat the curve is.

Fig. 5 Good rentention Chart

What is good API retention?

Day 1 should have the largest drop off. Most platforms have a set of developers that sign up, play around with the API in a sandbox, but never returns. Once a customer finds value, they keep consuming the API on Day 2, Day 3, and so on. These users found value in the API and stay integrated. On the other hand, if your API or SDK is buggy or doesn't provide immediate value, developers gradually remove their implementation causing retention to gradually drop down to 0%

Unlike the previous chart, this chart shows what bad API retention looks like. Notice how in this case, the retention goes all the way down to 0%.

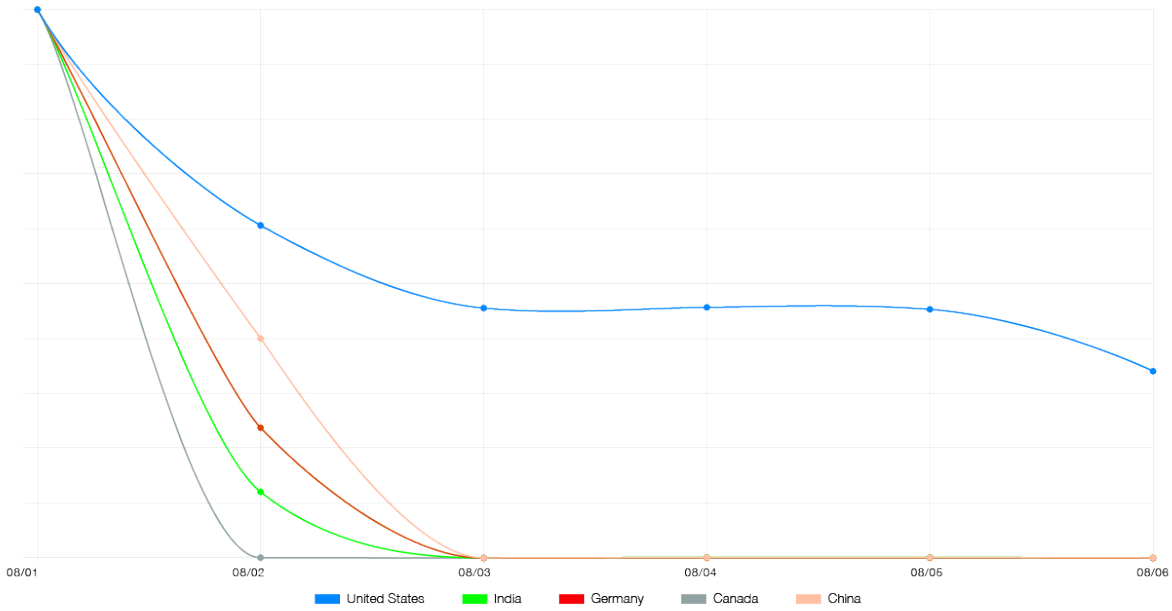
A product with a retention curve that continues to slope down until hitting 0% has poor retention. Whereas, a product with a retention curve that levels out or increases slightly has good retention.

Fig. 6 Poor Retention Chart

Copyright © 2020 moesif.inc

Breaking down by segments

Previously, we were looking at our entire user base. However, to dig deeper into why retention drops off, it's important to find the variants that drive this dip. One way to do this is by grouping your users into buckets. This can be broken down by user properties such as the country they are from, user acquisition channels, or even product specific information such as the SDK the developer was using.



Group	Total	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5
United States	26	100.00%	49.65%	43.017%	42.00%	41.71%	29.00%
India	28	100.00%	25.00%	0.00%	0.00%	0.00%	0.00%
Germany	38	100.00%	13.13%	0.00%	0.00%	0.00%	
Canada	34	100.00%	0.00%	0.00%	0.00%		
China	29	100.00%	0.00%	0.00%			

Fig. 7 Segmented retention chart

In this case, we gain more visibility into what's driving long term retention. While the overall retention curve was relatively flat (i.e. good retention), we see that the majority of our non-United States customers drop down to 0%. Only the United States users continue to leverage the API. We should further investigate why non US users don't use the API. Is it due to high latency? Non-localized documentation? Local laws and regulation preventing adoption? This might especially be true in financial, healthcare, and other regulated industries.

Copyright © 2020 moesif.inc

Defining initial and returning actions

From a business standpoint, the returning action should be something only active users perform and get value out of your product. To properly measure retention you need to define the initial action and a returning action. While not required, initial action usually is the first action a new user makes with your product.

Previously, we were just tracking new users who made their first API Call, and then made any returning API Call. However, we can get much more specific on what those initial or returning actions are. For example, we may consider a user active only if they returned and made a payment transaction on the API.

Let's say you're building a payments API that manages credit card transactions. The main value provided by the API is processing payments via credit card so a merchant gets paid, thus we consider an active user who processes at least one credit card payment via the API per day.

For an API, we may track retention as follows:

First Event Criteria: New user who made very first API Call
Returning Event Criteria: Came back and made at least one payment transaction

Conclusion

There are many deeper API metrics that you should be tracking if you intend to build a growing API program. However, if you haven't started taking a data-driven approach yet, these metrics are a good starting point. As you grow your API program, you'll be tracking other KPIs that can show you the health of your API program. Tools like Moesif API Analytics can help you get started measuring these metrics with just a quick SDK installation.

Copyright © 2020 moesif.inc